

FIG. 1

```
FUNCTION (  
    unsigned int in1,  
    unsigned int in2,  
    unsigned int in3,  
    unsigned int in4,  
    unsigned int in5,  
    unsigned int in6,  
    unsigned int select1,  
    unsigned int select2,  
    unsigned int *out  
)  
{  
    unsigned int m1, m2, m3, m4;  
  
    m1 = in1 + in2;  
    if (select1 == 0) {  
        m2 = in3 + in4;  
    }  
    else {  
        m2 = in3 + in5;  
    }  
    m3 = m1 + m2;  
  
    if (select2 == 0) {  
        m4 = m3 * in6;  
    } else {  
        m4 = m3 * m2;  
    }  
    *out = m4;  
}
```

FIG. 2

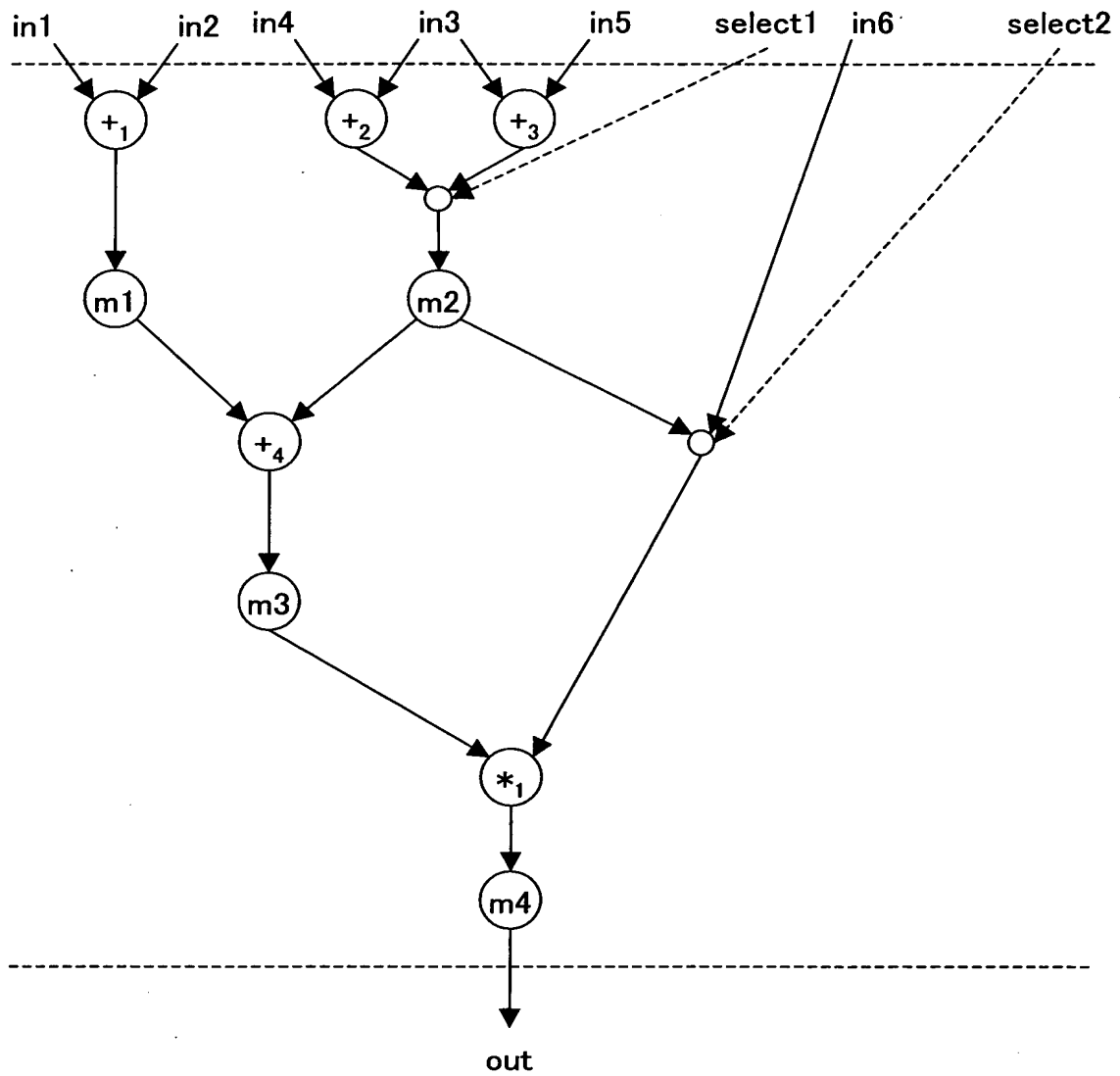


FIG. 3

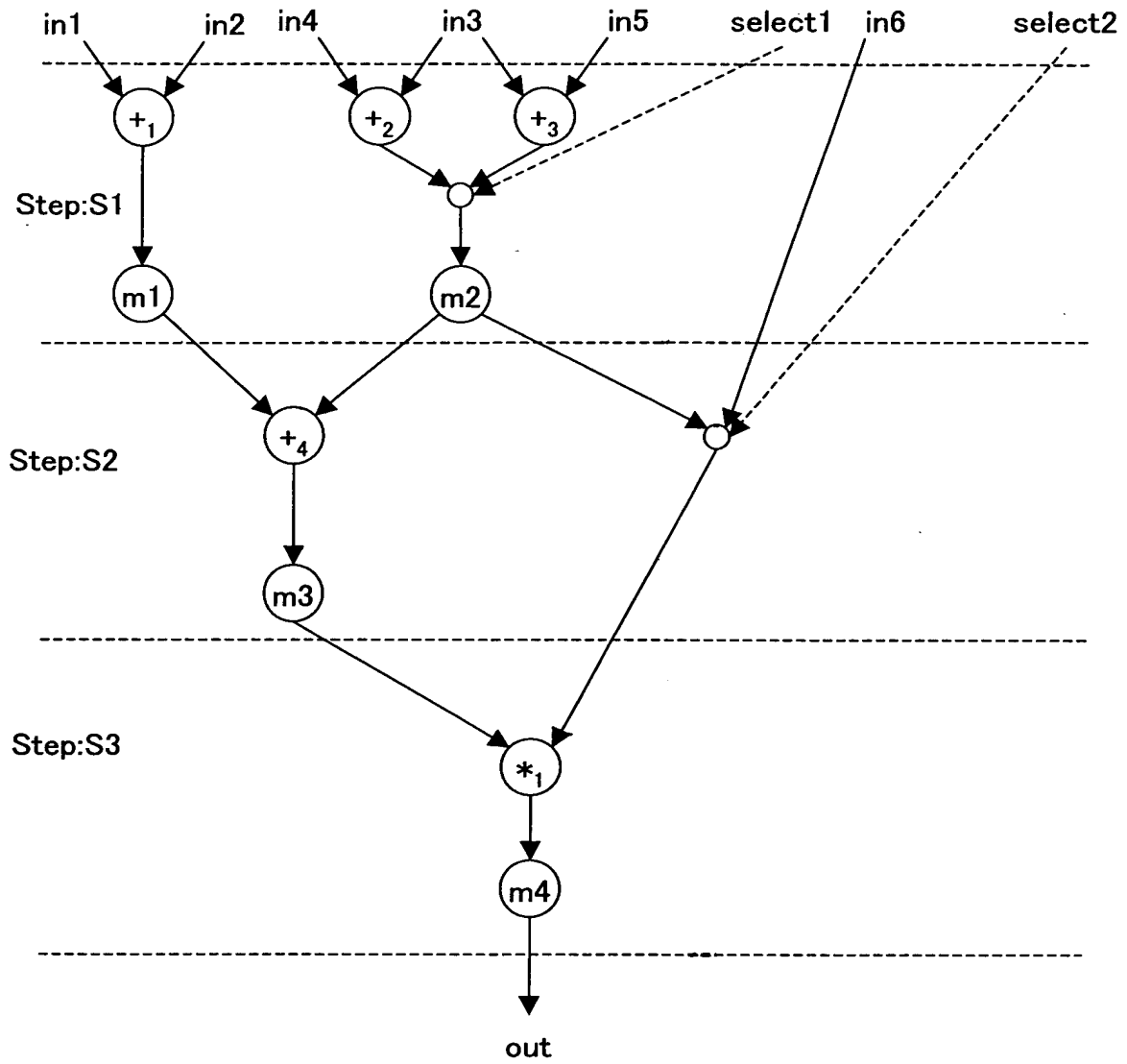


FIG. 4

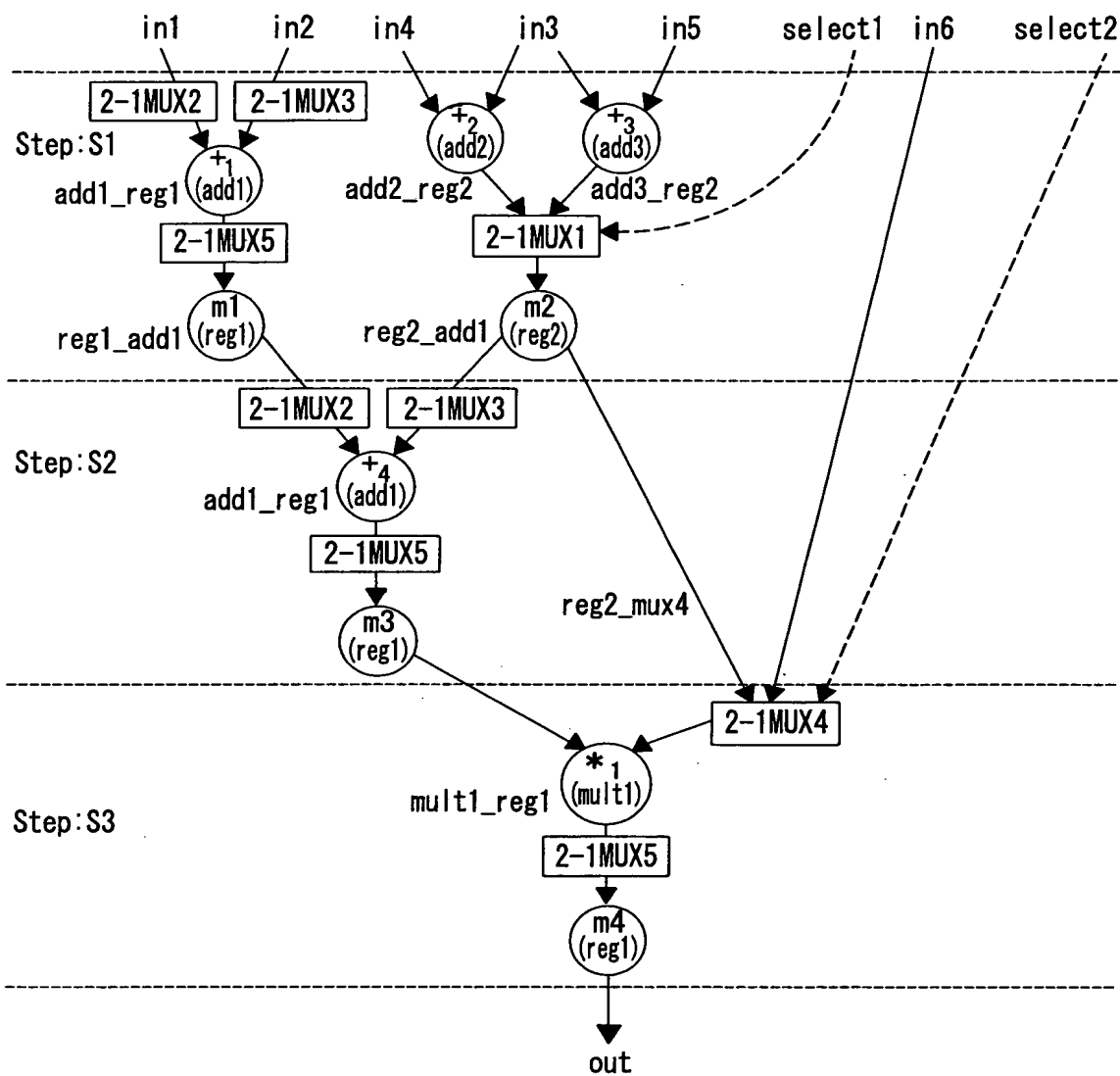


FIG. 5A

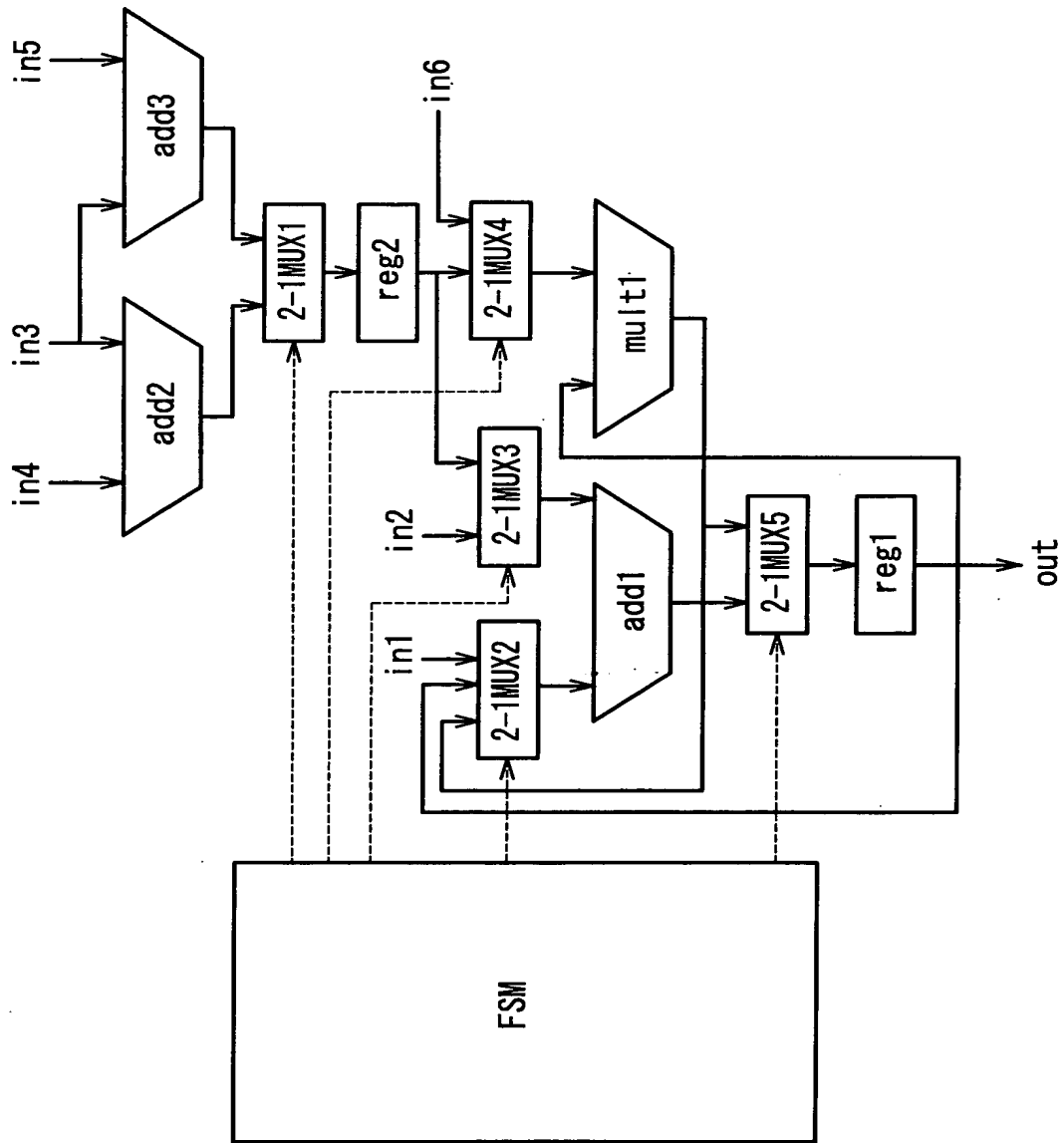


FIG. 5B

701

	S1	S2	S3
reg1	m1	m3	m4
reg2	m2		
add1	+ <sub>1</sub>	+ <sub>4</sub>	
add2	+ <sub>2</sub>		
add3	+ <sub>3</sub>		
mult1			* <sub>1</sub>
2-1MUX1	add2_reg2 add3_reg2		
2-1MUX2	in1	reg1_add1	
2-1MUX3	in2	reg2_add1	
2-1MUX4			reg2_mux4 in6
2-1MUX5	add1_reg1	add1_reg1	mult1_reg1

FIG. 6

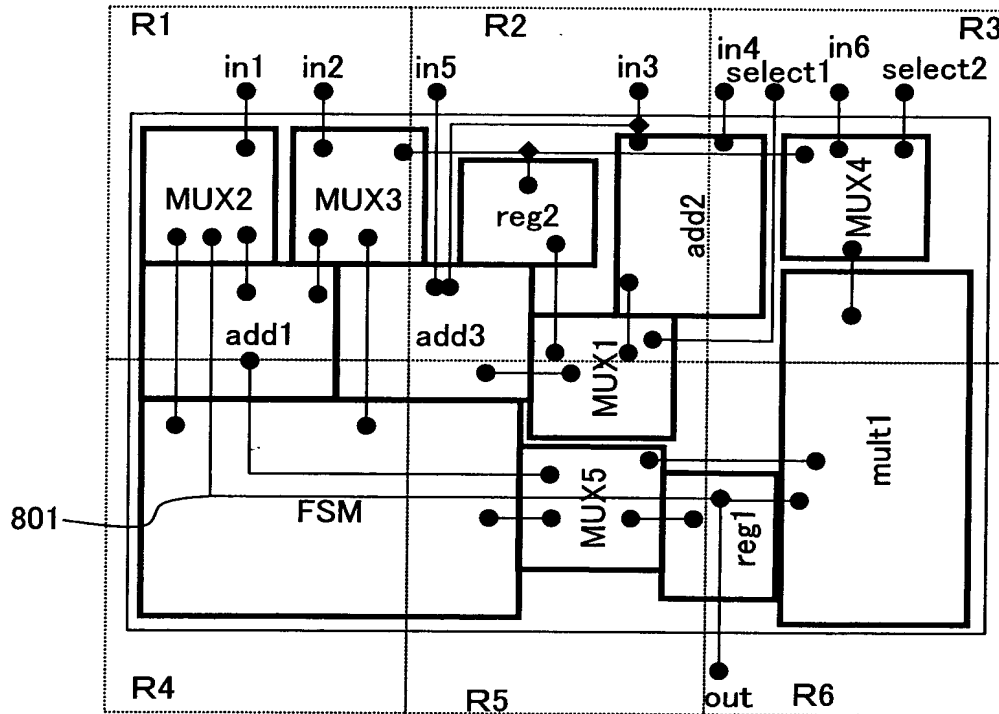


FIG. 7



	reg1	reg2	add1	add2	add3	2- 1MUX1	2- 1MUX2	2- 1MUX3	2- 1MUX4	2- 1MUX5	FSM
reg1		5	7	3	5	3	8	6	5	1	3
reg2			4	2	3	3	3	2	4	4	6
add1				6	2	4	1	2	8	6	3
add2					3	3	4	3	2	5	7
add3						2	3	2	5	3	3
2-1MUX1							5	4	5	2	3
2-1MUX2								2	4	5	3
2-1MUX3									4	5	3
2-1MUX4										6	8
2-1MUX5											3
FSM											

FIG. 8

R1	R2	R3	R4	R5	R6
9	6	6	4	6	2

FIG. 9

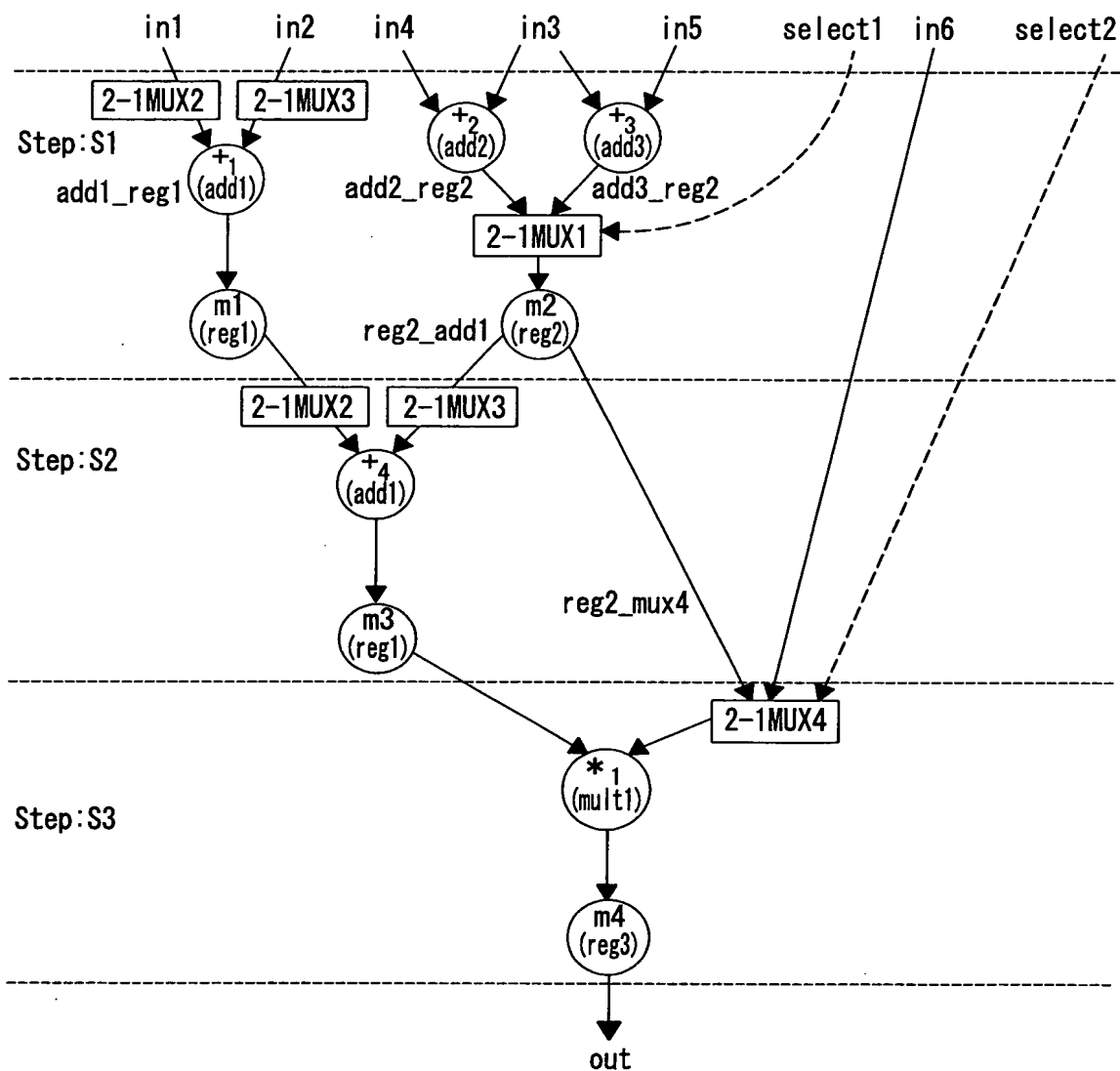


FIG. 10A

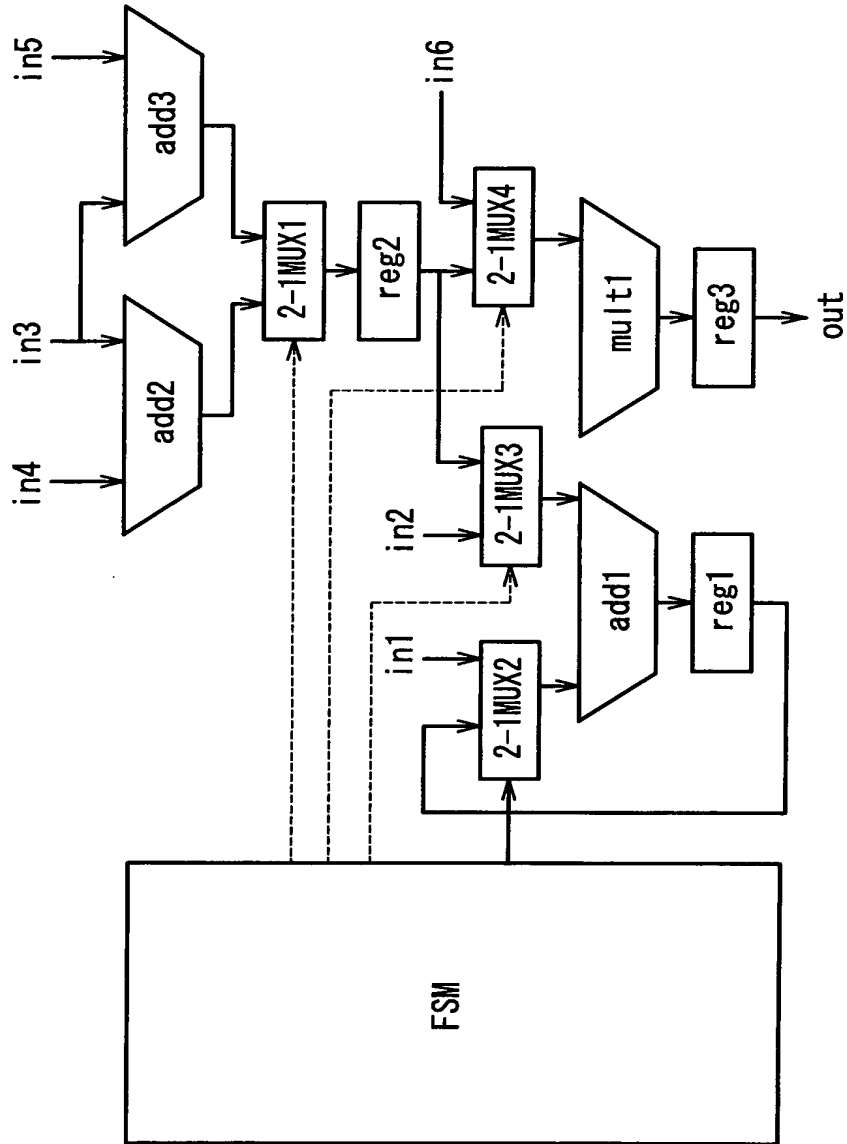


FIG. 10B

	S1	S2	S3
reg1	m1	m3	
reg2	m2		
reg3			m4
add1	+ <sub>1</sub>	+ <sub>4</sub>	
add2	+ <sub>2</sub>		
add3	+ <sub>3</sub>		
mutl1			* <sub>1</sub>
2-1MUX1	add2_reg2 add3_reg2		
2-1MUX2	in1	reg1_add1	
2-1MUX3	in2	reg2_add1	
2-1MUX4			reg2_mux4 in6

FIG. 11

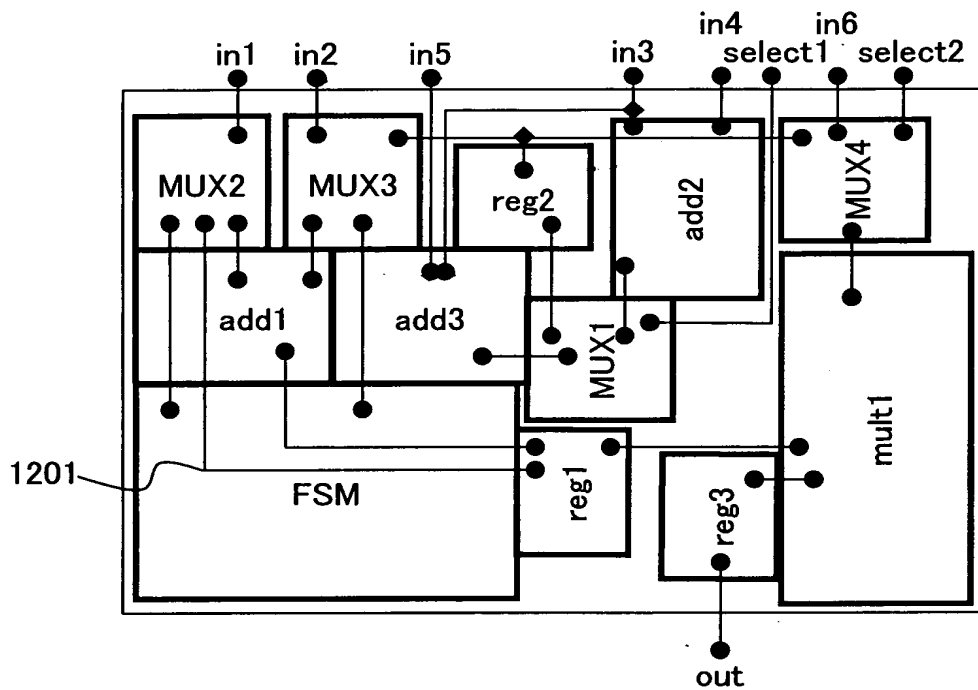


FIG. 12

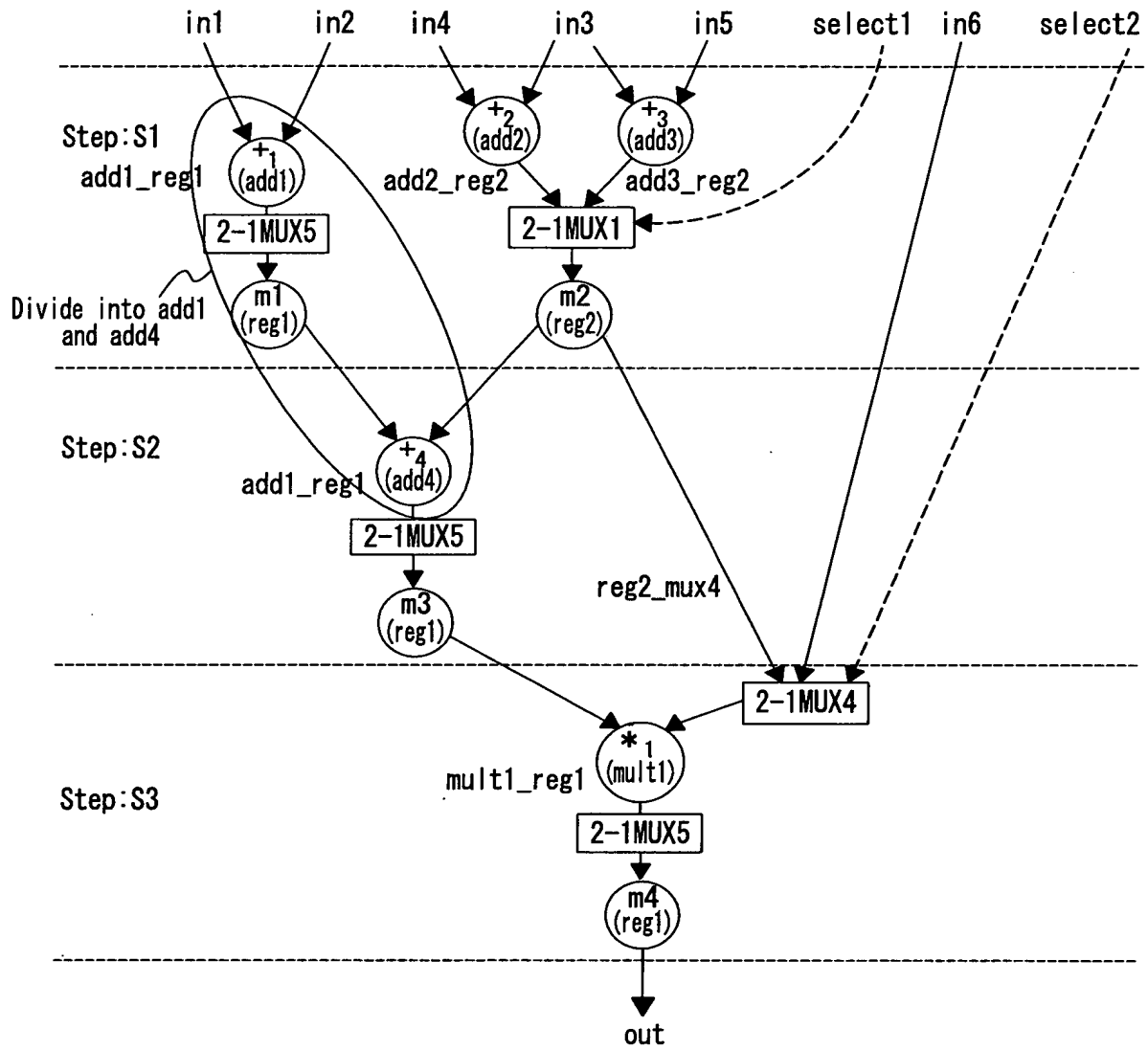


FIG. 13A

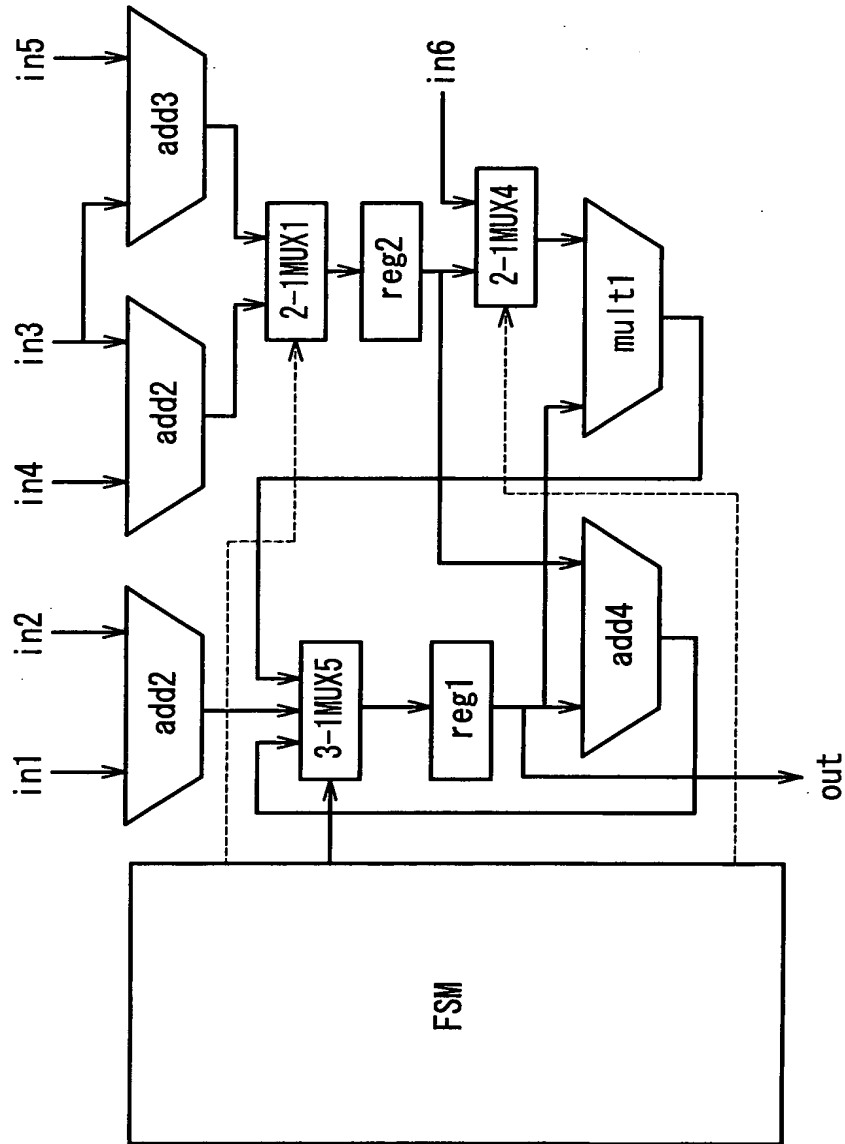


FIG. 13B



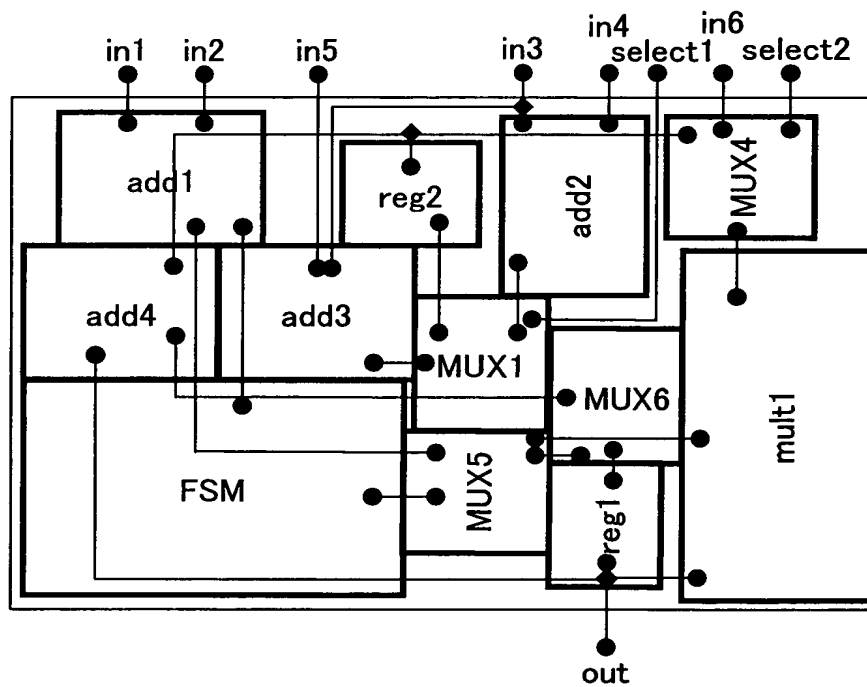


FIG. 14

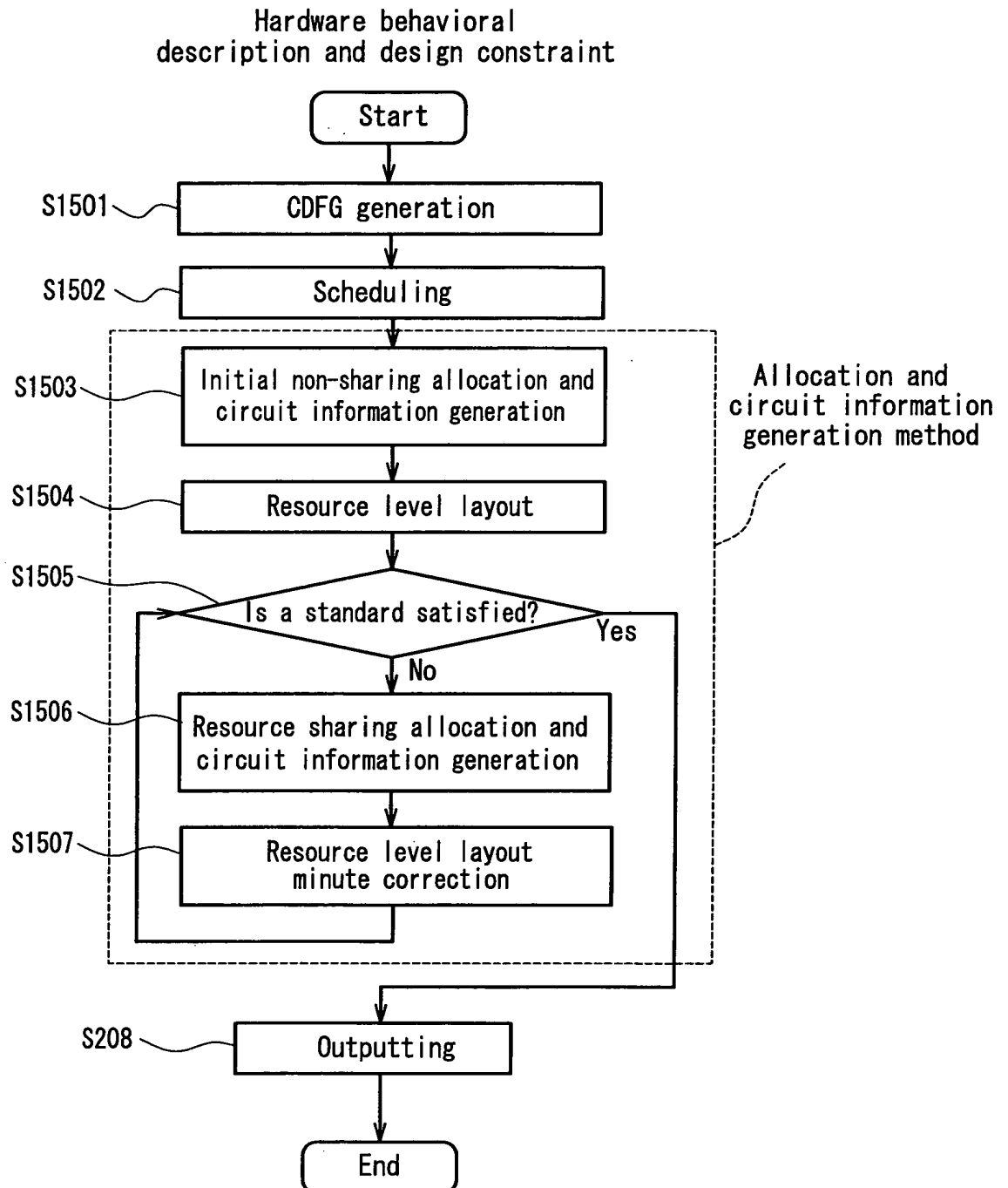


FIG. 15

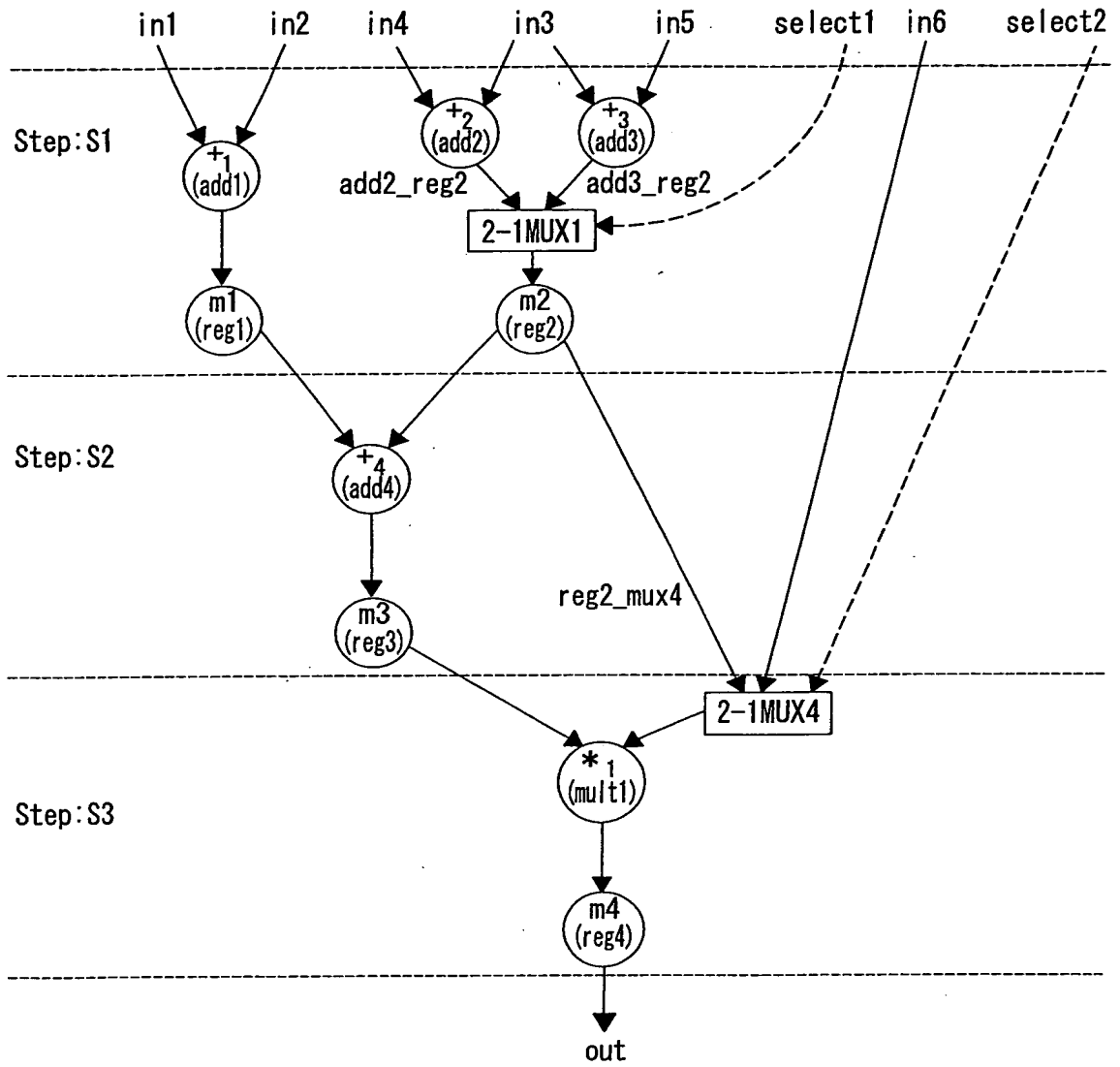


FIG. 16A

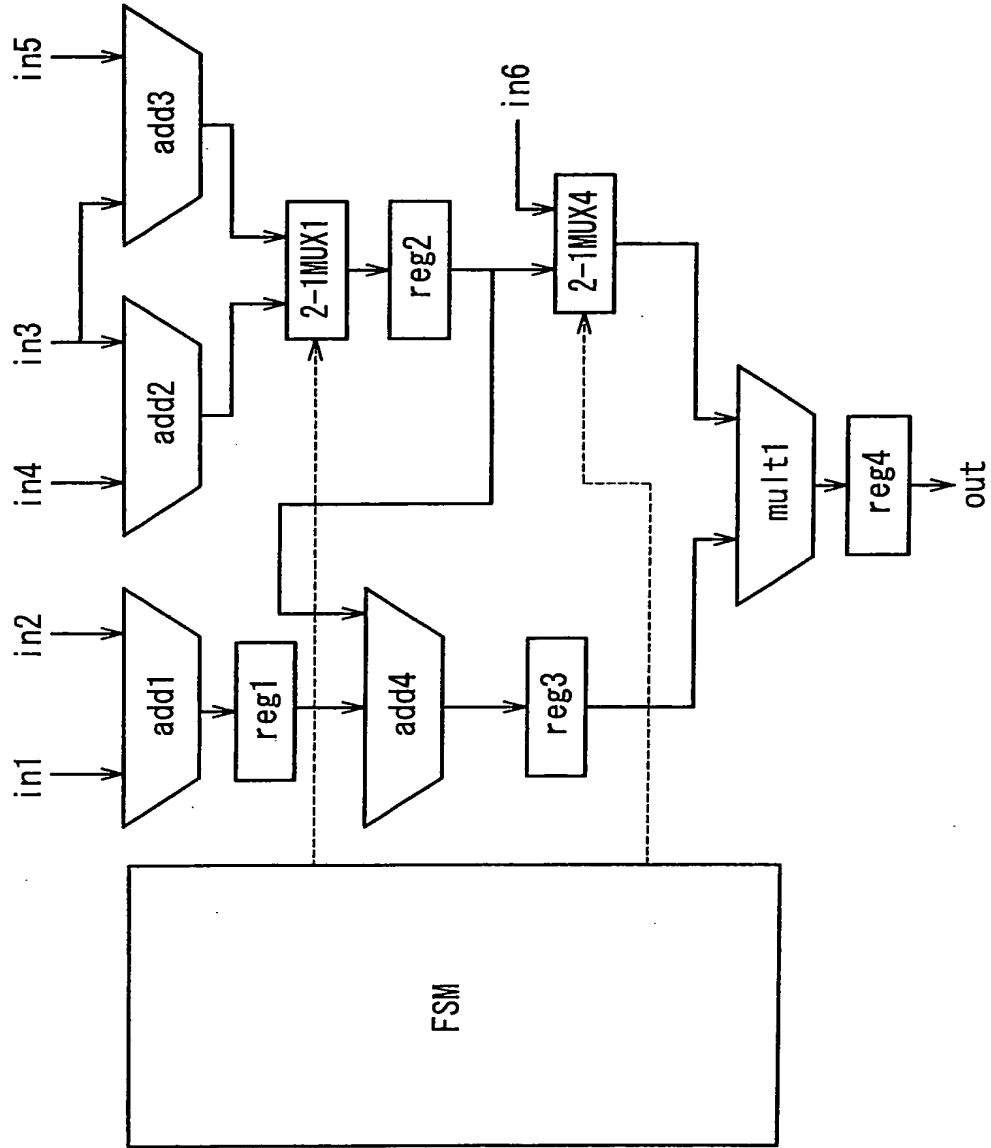


FIG. 16B

m1 or m2 can be shared with m3 and m4

	S1	S2	S3
reg1	m1		
reg2	m2		
reg3		m3	
reg4			m4
add1	+ <sub>1</sub>		
add2	+ <sub>2</sub>		
add3	+ <sub>3</sub>		
add4		+ <sub>4</sub>	
mutl1			* <sub>1</sub>
2-1MUX1	add2_reg2 add3_reg2		
2-1MUX2			reg2_mux4 in6

FIG. 17

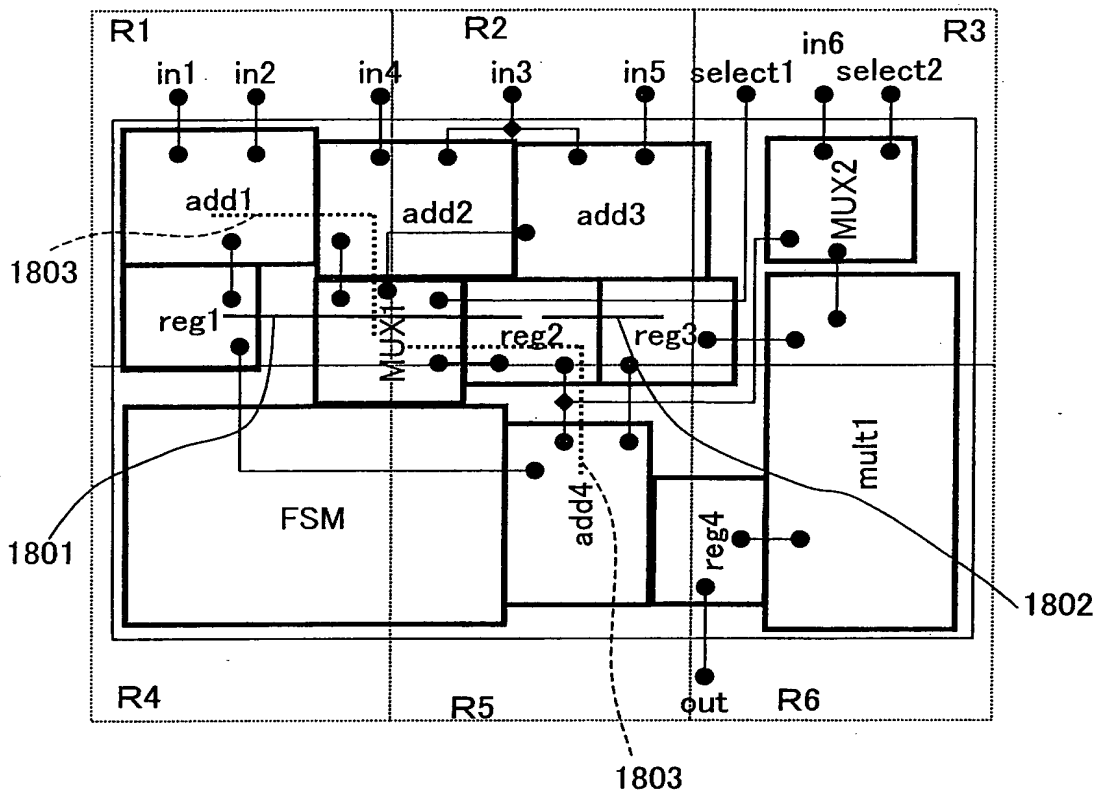


FIG. 18

	reg1	reg2	reg3	reg4	add1	add2	add3	add4	mult1	2-1MUX1	2-1MUX2	FSM
reg1		4	5	9	1	3	5	6	7	2	8	2
reg2			1	5	5	2	2	4	3	1	4	4
reg3				2	6	2	1	2	2	2	2	4
reg4					9	4	3	1	1	4	4	4
add1						1	3	6	10	3	8	3
add2							1	4	5	1	3	3
add3								3	4	2	2	4
add4									2	3	5	2
mult1										4	2	4
2-1MUX1											4	1
2-1MUX1												6
FSM												

FIG. 19

R1	R2	R3	R4	R5	R6
7	7	6	1	3	2

FIG. 20



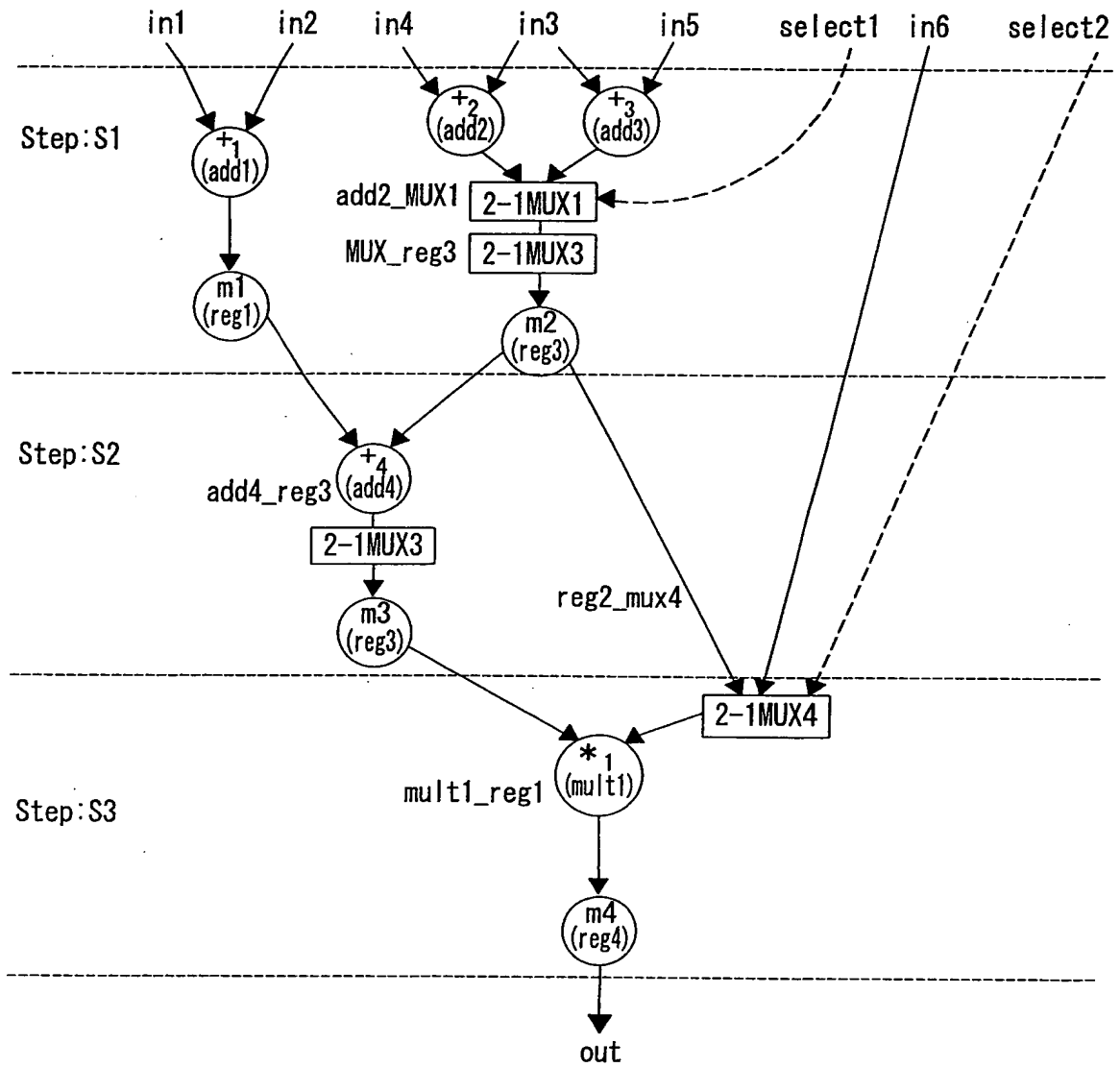


FIG. 21A

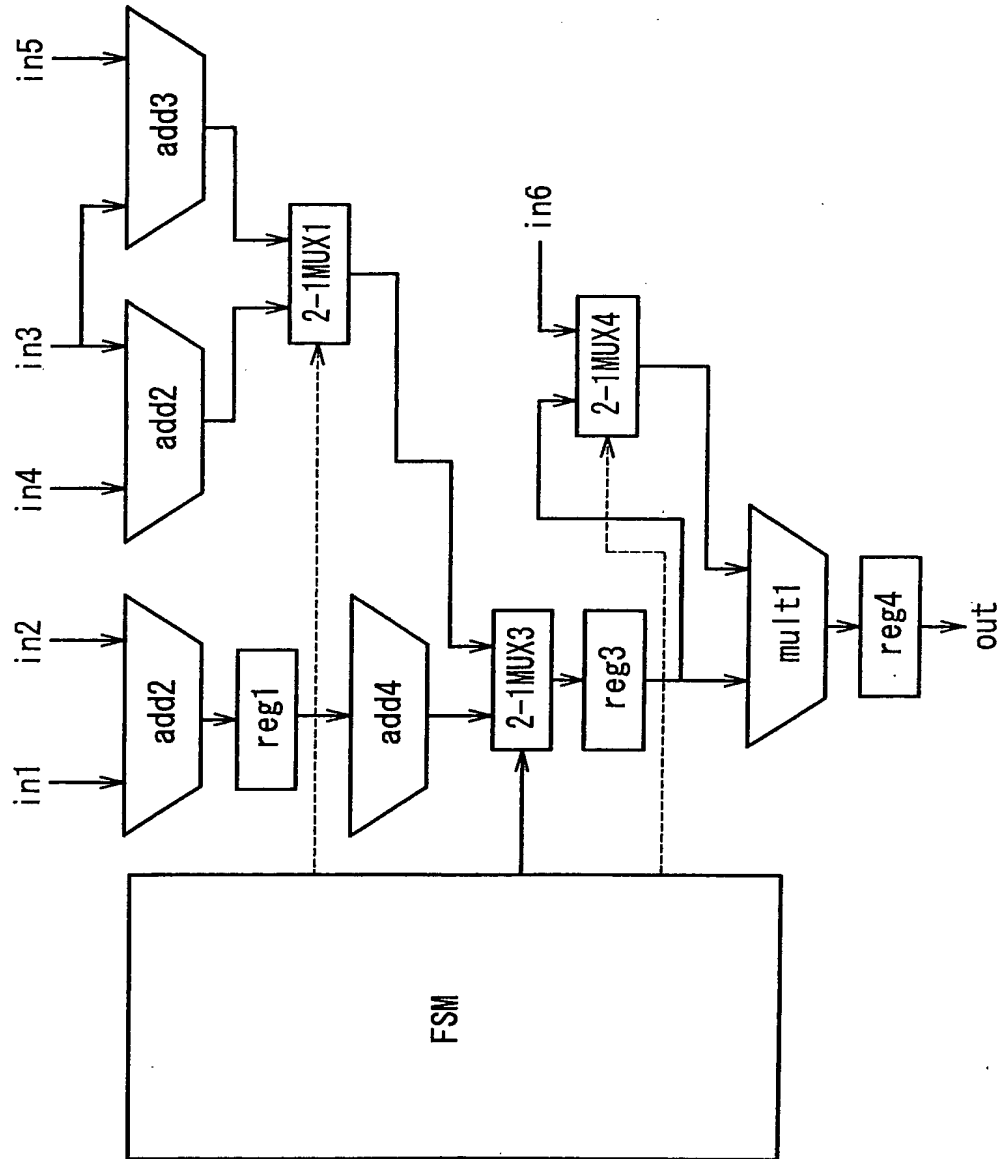


FIG. 21B

m2 and m3 are shared

	S1	S2	S3
reg1	m1		
reg3	m2	m3	
reg4			m4
add1	+ <sub>1</sub>		
add2	+ <sub>2</sub>		
add3	+ <sub>3</sub>		
add4		+ <sub>4</sub>	
mutl1			* <sub>3</sub>
2-1MUX1	add2_reg2 add3_reg2		
2-1MUX2			reg2_mux4 in6

FIG. 22

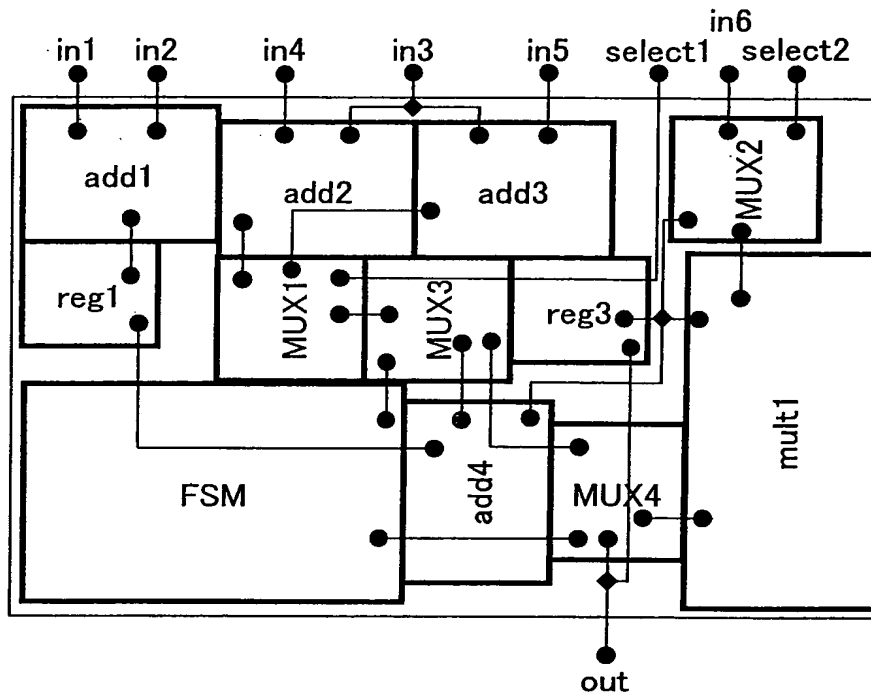


FIG. 23

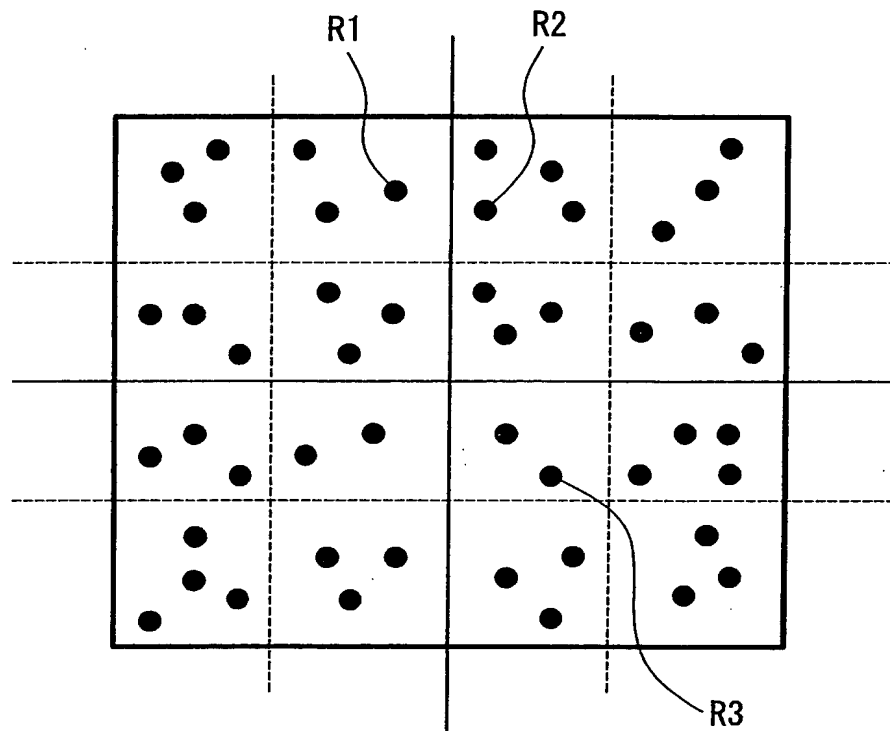


FIG. 24

# Hardware behavioral description and design constraint

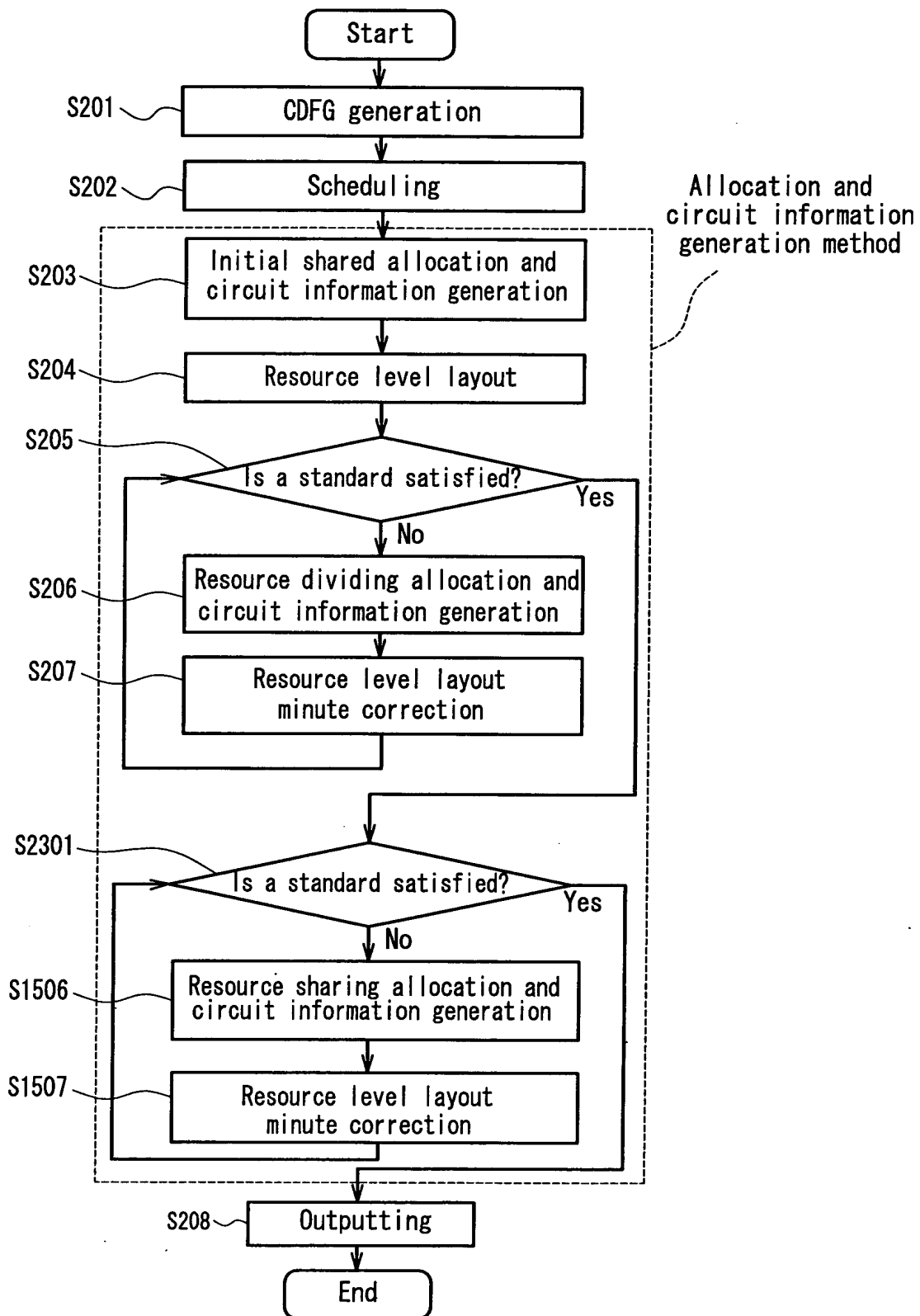


FIG. 25

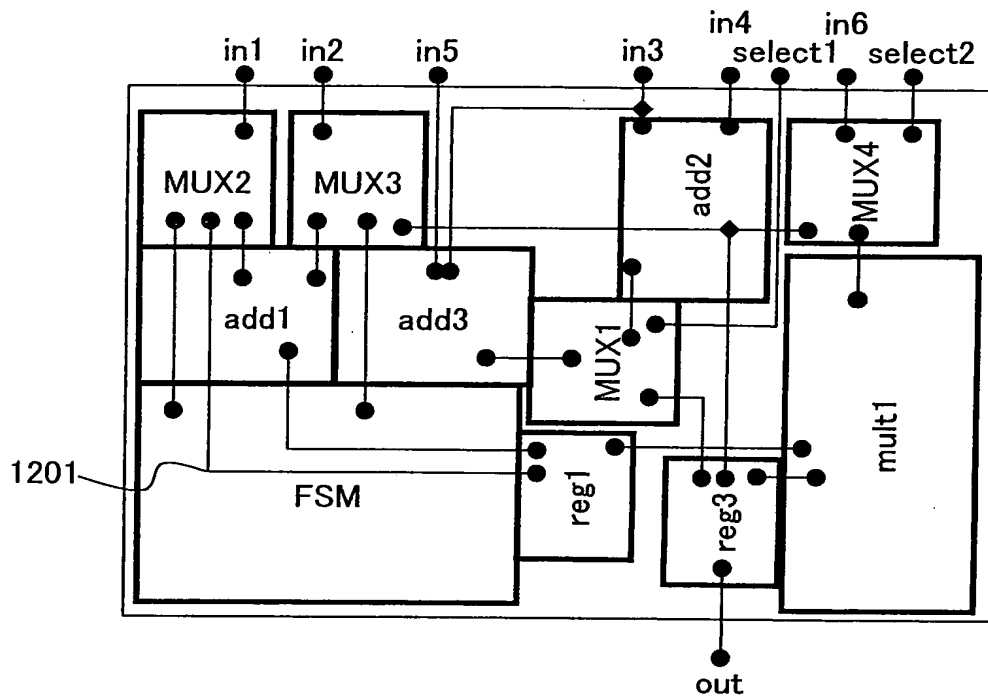


FIG. 26

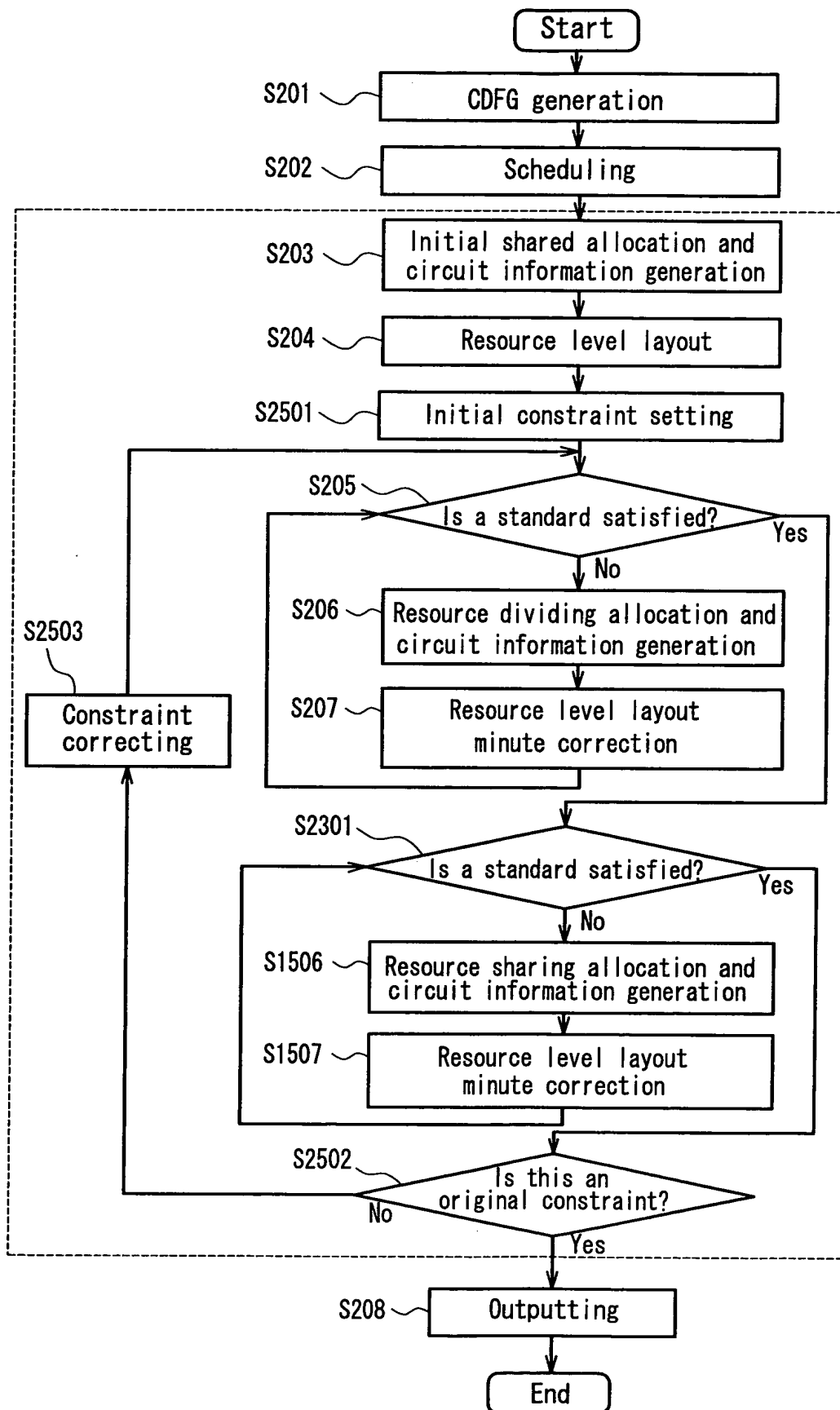


FIG. 27



# Hardware behavioral description and design constraint

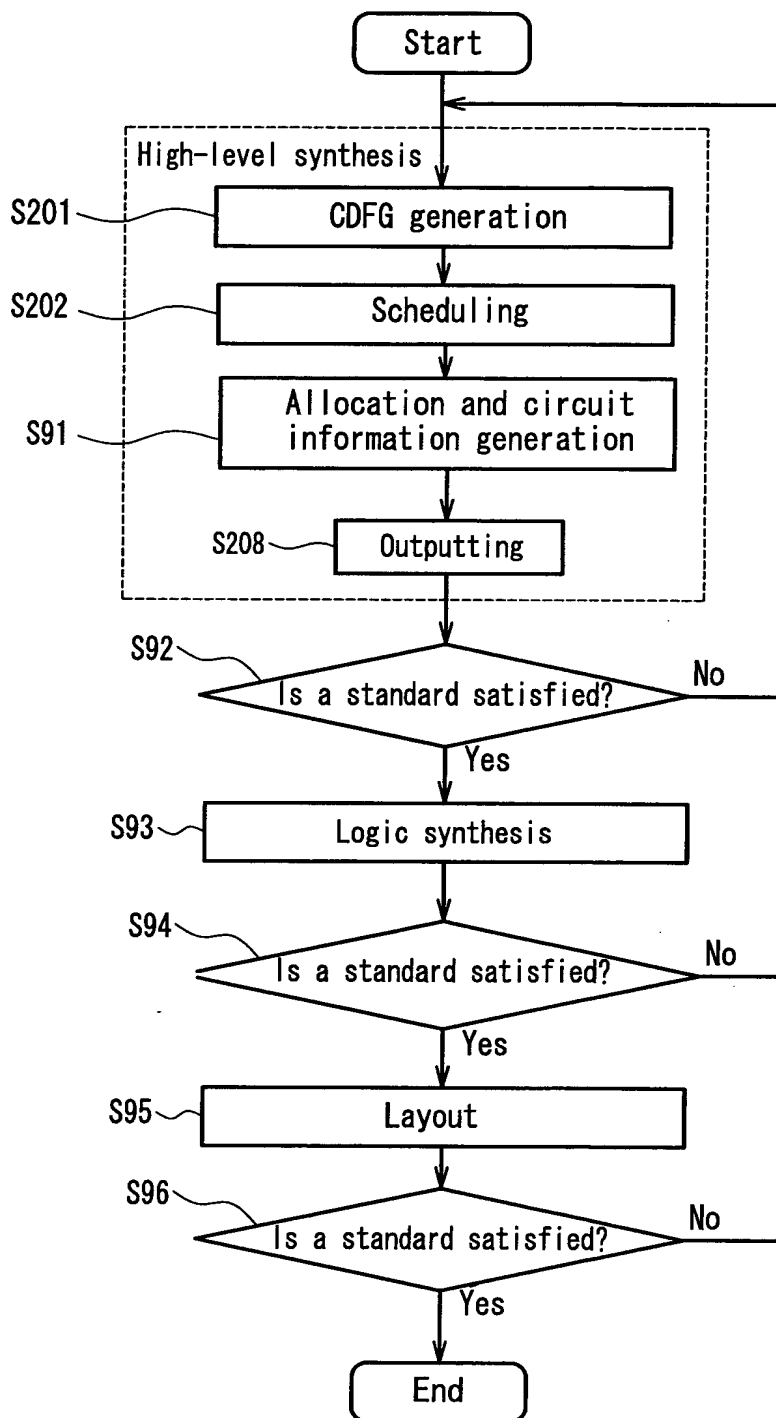


FIG. 28  
PRIOR ART